

RESPOSTA: Na chamada do procedimento que recebe o parâmetro/argumento por valor, o valor armazenado na variável não é alterado. Já na função que recebe por referência, o valor é alterado, passando a ser o dobro do valor anterior inicial. Porque na versão por referência, como o próprio nome indica, é passado como parâmetro o endereço de memória real da variável, então dentro da função “sabendo-se” esse endereço, o valor da variável pode ser modificado de dentro da função, enquanto na passagem por valor é criada uma cópia da variável, e quaisquer modificação altera esta cópia, sem acesso à variável original e seu respectivo endereço de memória real. Na estratégia de passagem de parâmetros por valor, é criada uma cópia da variável que possui um outro endereço de memória, e qualquer alteração modifica apenas o valor desta cópia criada, mas não muda nada na variável original, uma vez que a função não possui “acesso” ao endereço de memória desta. Já na estratégia de passar parâmetros por referência, a função “sabe” o endereço de memória da variável que foi recebida como argumento, então qualquer modificação altera realmente o valor da variável original, pois é como se fosse a própria variável em si que estivesse sendo recebida pela função, pois não há cópia, mas sim o recebimento do verdadeiro endereço de memória do parâmetro.